10

Using Scripts (Macros) in MIPAV

In this chapter . . .

MIPAV provides three different methods of customizing the program. The first method involves developing scripts, which you use directly within the program (see "Developing and using scripts"). In the second method, you also develop scripts but you initiate, or call, them from another program. Developing scripts does not require programming skills or learning a new script language, and calling them from another program may require, depending on the program, only minimal programming knowledge ("Combining scripts and other programs" on page 447). This chapter discusses both of these methods of developing and using scripts.

The third method of customizing MIPAV—developing plug-in programs—does require Java programming skills. For more information about this method, refer to Chapter 11, "Developing Plug-in Programs."

Developing and using scripts

Scripts, sometimes referred to as *macros* in other programs, record a series of commands or actions on specific images or groups of images that you can run with a single command. Using scripts can increase productivity and improve efficiency in performing commonly repeated actions or a series of actions. Most important, you can use scripts to process a large set of user-defined images.

This section explains the following tasks:

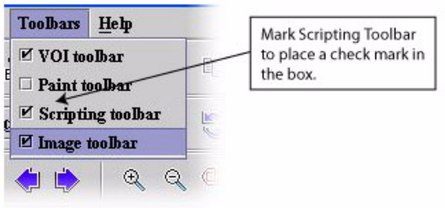
- Setting up scripting (refer to "Setting up scripting" on page 425
- Planning scripts (refer to "Planning scripts" on page 430)
- Creating scripts (refer to "Creating scripts" on page 433)
- Running scripts (refer to "Running scripts" on page 440)
- Editing and deleting scripts (refer to "Editing and deleting scripts" on page 443)

Setting up scripting

Before you can create, edit, run, or delete scripts, you need to first display the scripting toolbar and then set up scripting by selecting the scripts home directory.

To display the scripting toolbar

Like the paint toolbar, you can choose to show it on an as-needed basis or to show it on an as-needed basis. To show the scripting toolbar on an as-needed basis, select Toolbar > Scripting Toolbar (Figure 280) in the MIPAV window.

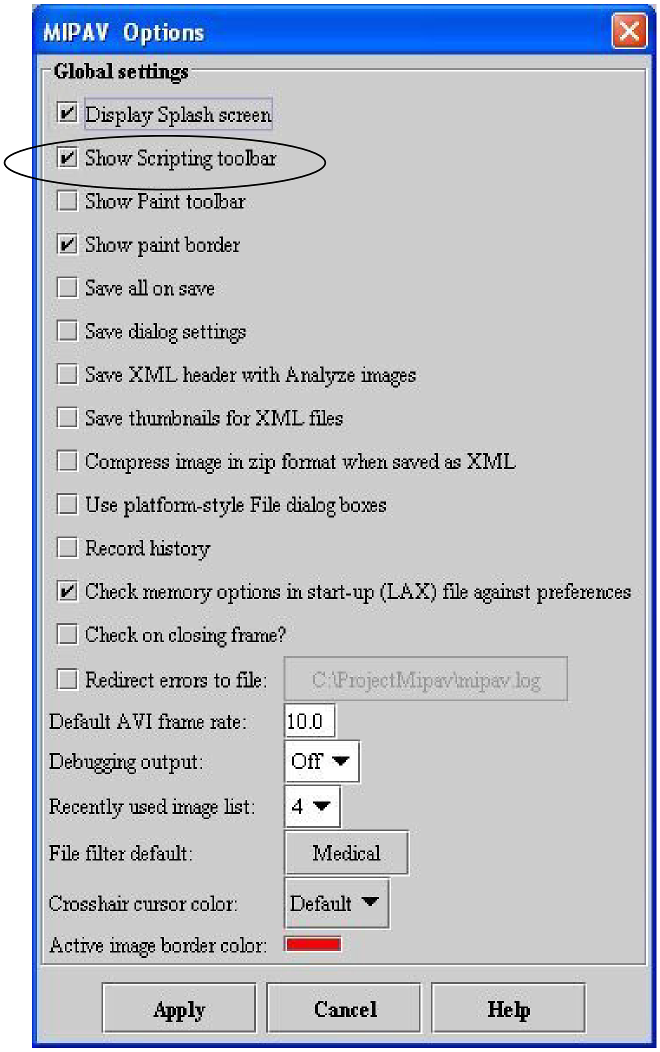


Mark Scripting Toolbar to place a check mark in the box.

Figure 280. Scripting Toolbar command on the Toolbar menu

The scripting toolbar immediately appears beneath the VOI toolbar or, if the Paint toolbar is also displayed, immediately below the Paint toolbar.

To always show the scripting toolbar, select Help > Program Options in the main MIPAV window and then select Show scripting toolbar in the MIPAV Options dialog box (Figure 281 on page 426).



MIPAV Options

Global settings

☑ Display Splash screen

☑ Show Scripting toolbar

□ Show Paint toolbar

☑ Show paint border

□ Save all on save

□ Save dialog settings

□ Save XML header with Analyze images

□ Save thumbnails for XML files

□ Compress image in zip format when saved as XML

□ Use platform-style File dialog boxes

□ Record history

☑ Check memory options in start-up (LAX) file against preferences

□ Check on closing frame?

□ Redirect errors to file: C:\ProjectMipav\mipav.log

Default AVI frame rate: 10.0

Debugging output: Off

Recently used image list: 4

File filter default: Medical

Crosshair cursor color: Default

Active image border color: 🟥

Figure 281. MIPAV Options dialog box with Show scripting toolbar selected

To set up scripting

The *scripts home* is the directory in which you store scripts and from which you run them. After you select a scripts home, the name of one of the scripts (which are ordered alphabetically) in that directory appears after Current Script on the scripting toolbar (Figure 281). If there is more than one script in the directory, Current Script becomes a list box from which you can select the script you want to run.

1 Start MIPAV. The initial MIPAV window opens.

2 Open an image. The expanded MIPAV window appears.

3 Select Toolbars > Scripting if the scripting toolbar is not displayed. The scripting toolbar appears.

Notice that a dimmed rectangle appears immediately following the words *Current Script* in the scripting toolbar (Figure 282).

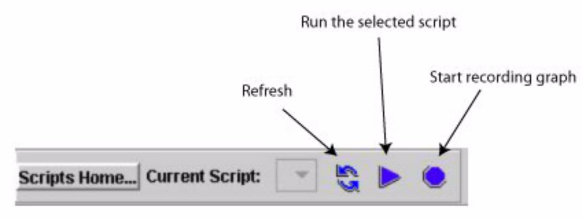
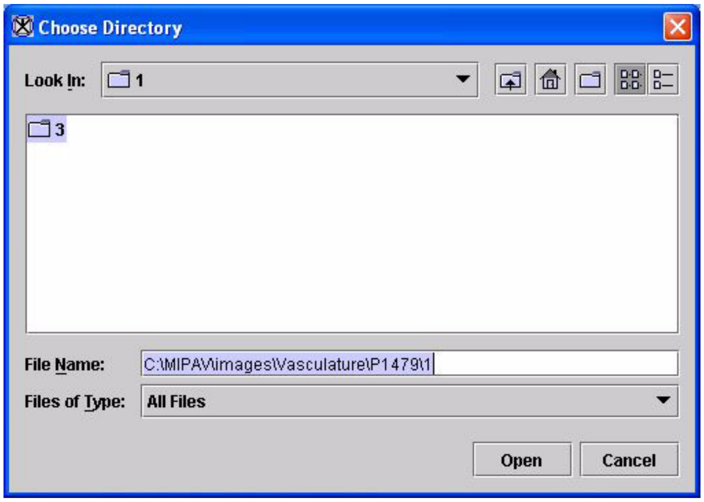


Figure 282. Initial Scripting toolbar

4 Click Scripts Home. The Choose Directory dialog box (Figure 283) appears.

Note: The Choose Directory dialog box only shows directories, not individual files.



Look In: 1

3

File Name: C:\MIPAV\images\Vasculature\P1479\1

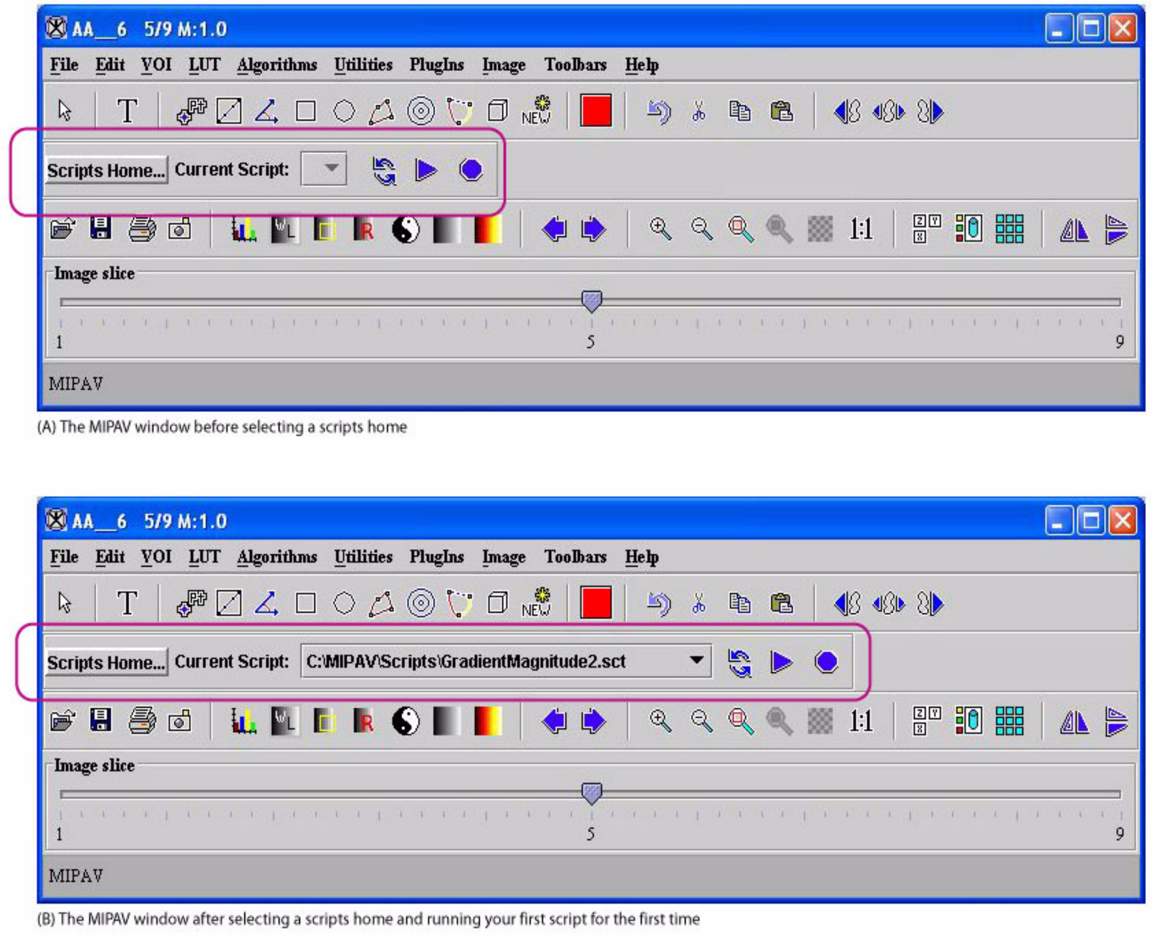
Files of Type: All Files

Figure 283. Choose Directory dialog box, which shows only directories, not individual files

5 Select a directory in which to store your scripts.

6 Click Open. The Choose Directory dialog box closes.

After you select a scripts home and run a script for the first time, Current Script becomes a list box that shows the name of a script in the scripts home directory (Figure 284B). After creating more scripts, you can select the one you want to run from Current Script.


AA__6   5/9 M:1.0
File Edit VOI LUT Algorithms Utilities PlugIns Image Toolbars Help
Scripts Home...
Current Script:
Image slice
1
5
9
MIPAV

(A) The MIPAV window before selecting a scripts home

AA__6   5/9 M:1.0
File Edit VOI LUT Algorithms Utilities PlugIns Image Toolbars Help
Scripts Home...
Image slice
1
5
9
MIPAV

Current Script: C:\MIPAV\Scripts\GradientMagnitude2.sct

(B) The MIPAV window after selecting a scripts home and running your first script for the first time

Figure 284. The MIPAV window (A) before selecting a scripts home and (B) after selecting a scripts home and running the first script

Planning scripts

Although MIPAV allows you to easily create scripts, it is important to plan scripts before creating them, particularly if the scripts perform actions on VOIs. First, however, you need to understand:

- Methods for creating scripts
- Actions that can be performed in scripts
- Use of VOIs in scripts

METHODS FOR CREATING SCRIPTS

There are two methods for creating scripts: the active mode and the group mode.

Active mode

In active mode, one or more images are already open when you create the script. Because images are open, MIPAV assumes that the script applies to the currently active image. The *active image* is the image that is currently selected (Figure 285A).

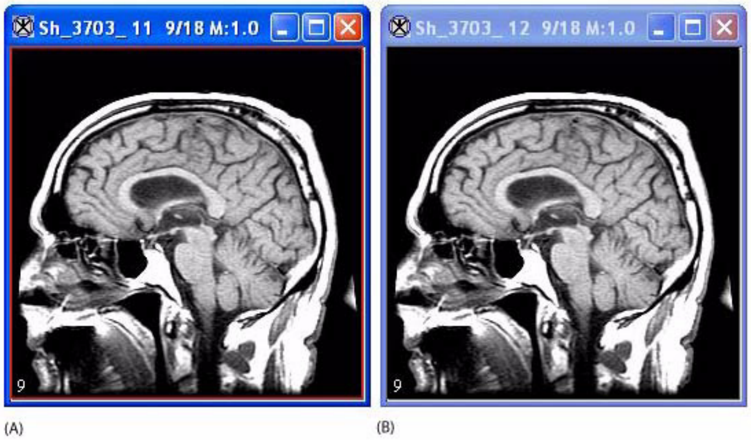


Figure 285. An (A) active image, whose title bar is highlighted, and (B) inactive image, whose title bar is dimmed

To create a script in active mode, you simply click ●, the Start Recording Script icon, on the scripting toolbar.

Group mode

In group mode, no images are open when you begin creating the script, and the MIPAV window appears in its initial small size, which only displays some of the menus. Generally, you use the group mode to create scripts that apply to multiple image files.

To create a script in group mode, click File > Scripts > Record Script (Figure 286) in the MIPAV window.

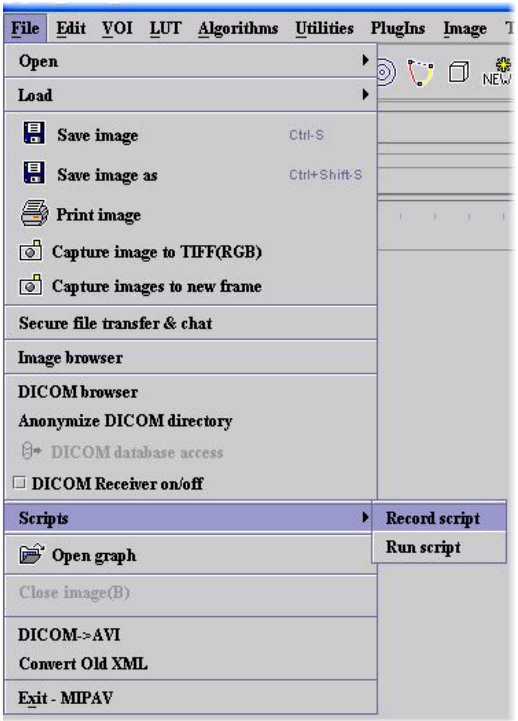


Figure 286. Record Script command on the File menu

Actions permitted in scripts

It’s also important to understand which actions or events you can record in scripts. For example, scripts do not record views, such as lightbox, triplanar, and volume renderer; nor do they record other visualization commands, such as those that adjust the lookup table.

Actions that you can record in scripts include:

- Opening, saving, and closing images
- Opening VOIs
- Performing algorithms on images or on open VOIs on images
- Applying utilities to images or to open VOIs on images
- Saving the global data page in the Output window
- Exiting from the MIPAV program (this action is only permissable when running scripts from other programs)

Caution: Do **not** use the Exit command in MIPAV scripts unless you are calling the script from another program or scripts (Perl, C++, Windows batch).

Use of VOIs in scripts

The *only* VOI commands that can be recorded in scripts are the open VOI commands:

- VOI > Open VOI
- VOI > Open All VOIs
- VOI > Open VOIs from

What this means is that, if you create and save VOIs while you are recording a script, the script does not record these VOIs and does not apply them to the images on which you want to run the script. Therefore, to apply VOIs to images in scripts, you need to:

- Create and save the VOIs in *advance* of creating a script

- Open and apply the VOIs when you are recording the script

After you create VOIs, you can save the VOIs through the Save VOI commands in the VOI menu in the MIPAV window:

- VOI > Save VOI
- VOI > Save VOI as
- VOI > Save All VOIs
- VOI > Save All VOIs to

Saving VOIs: Some Reminders

You can save VOIs under names that MIPAV assigns or under names that you assign. If the VOIs have never been saved, MIPAV assigns names to VOIs when you use the Save VOI or Save All VOIs commands. The name of the first VOI or set of VOIs that you save is *Area1.voi*; the second is *Area2.voi*; and so on.

If you want to specify a different name for the VOI or set of VOIs: select VOI > Save VOI as or VOI > Save All VOIs to. The Save VOI as dialog box opens. Type a name for the VOI in File name and the extension .voi or .xml, and click Save. The program saves the file under the specified name.

Creating scripts

To create scripts using the active mode: groups of images

1 Start MIPAV. The initial MIPAV window and the Output window open.

2 Select File > Scripts > Run

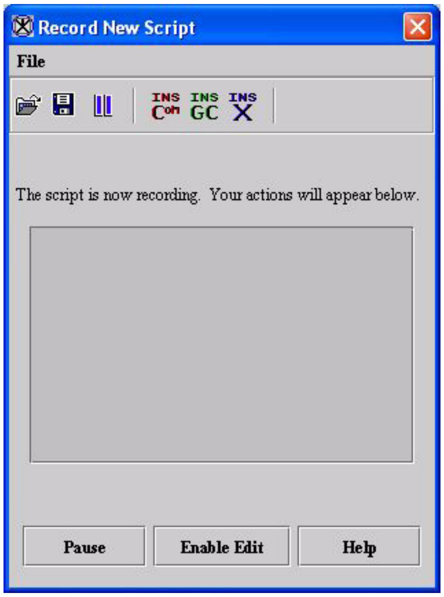
3 Open the images on which you want to run the script. The MIPAV window expands.

4 Click ●, the Start recording script icon, on the scripting toolbar to create a script in active mode.

Note: The icon turns red (●) to indicate that MIPAV is now recording a script.

The Record New Script dialog box (Figure 289) opens. The following appears at the top of the dialog box: “The script is now recording. Your actions will appear below.”

File	**Open**—Opens a previously saved script. When you select this command, the Open dialog box appears.
	Save—Saves the script under the specified name. When you select this command, the Save dialog box opens.
	Exit—Closes this dialog box without saving the script.
Open script	**Open**—Opens a previously saved script. When you select this command, the Open dialog box appears.
Save script	**Save**—Saves the script under the specified name. When you select this command, the Save dialog box opens.
Pause scripting	Stops recording the script. When you select this icon, the icon changes to ●, the Resume scripting icon.
Resume scripting	Activates the recording process again. When you select this icon, the icon changes to ⏸, the Pause scripting icon.



The script is now recording. Your actions will appear below.

Figure 288. Record New Script dialog box

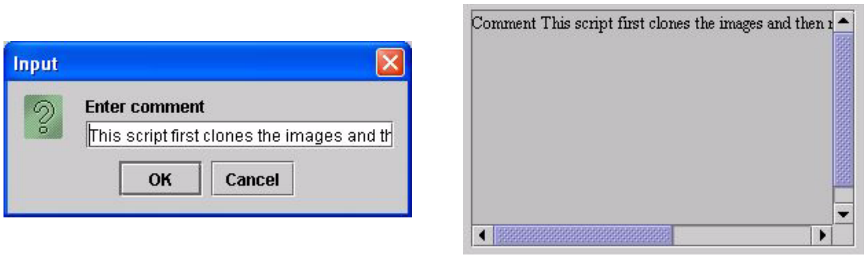
INS Com **Insert comment**	Allows you to insert a comment into the script. When you select this icon, the Input dialog box opens. Type the comment in Enter comment and click OK. MIPAV adds your comment into the script after the word *Comment*.  Enter comment: This script first clones the images and th Comment This script first clones the images and then r **Figure 287. Input dialog box (A) showing a comment and (B) the comment as it appears in the scripting box**
INS GC **Insert command to collect garbage (free memory)**	Frees memory by inserting the command to collect garbage, a Java method that clears all unnecessarily reserved memory. For more information, refer to “Managing memory resources” on page 97.
INS X **Insert command to end MIPAV**	Inserts the command for exiting MIPAV (**Exit**) into the script, which allows MIPAV to close as a part of the script. **Caution**: Do **not** use the Exit command in MIPAV scripts unless you are calling the script from another program or scripts (Perl, C++, Windows batch).
Scripting box	Displays the action commands (algorithms and utilities) and any comments in the script.
Pause	Stops recording the script. When you select this button, the name of the button changes to “Resume.”
Enable Edit	Allows you to make changes or corrections to the script. When you select this button, the scripting box turns from gray to white to indicate that you can now type, copy, paste, or delete information from the script, and the button name changes to “Disable Edit.”
Disable Edit	Prevents any changes or corrections being made to the script from typing, pasting, inserting, or deleting information. It does, however, allow MIPAV to record actions on images. When you select this button, the scripting box turns from white to gray, and the button name changes to “Enable Edit.”

Figure 288. Record New Script dialog box (continued)

Help	Displays online help for this dialog box.

Figure 288. Record New Script dialog box (continued)

5 Open any previously created VOIs that you want to use with the images.

6 Perform any algorithm or utility.

7 Repeat step 5 and step 6 as needed.

Note: When you are creating a script in active mode, the word *$active* appears in the commands recorded in the scripting box. For example, if you flip the image horizontally, the program adds the command **Flip $active1 Y** (refer to the example in "An Example: Recording a Script in Active Mode" on page 437).

8 Save and close the images.

9 Select File > Save in the Record New Script dialog box when you are finished creating the script. The Save dialog box opens.

10 Type a name for the script and either the .sct or .xml extension in File name.

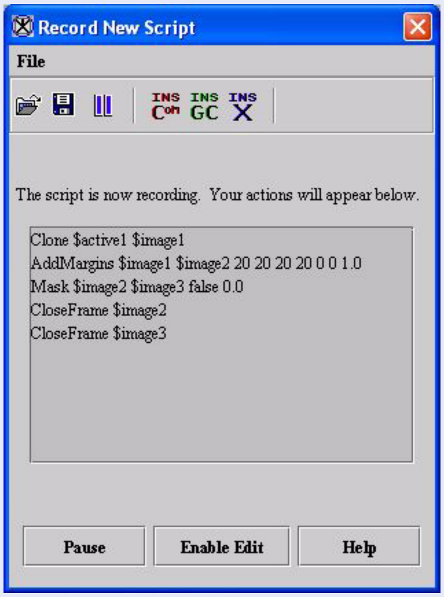
11 Click Save. The program saves the script in the scripts home directory.

12 Click File > Exit to close the Record New Script dialog box. The dialog box closes. You should now be able to select the script in Current Script.

An Example: Recording a Script in Active Mode

In this example, the user wants to clone an active image. The command in the script is "**Clone $active1 $image1**); in other words, clone the current image (**$active1**) to produce a copy of that image, which MIPAV labels "**$image1**". In the second action, the user wants to add margins using the Add Image Margins utility to the cloned image (**$image1**). This action produces another image, **$image2**.

On **$image2**, the user wants to apply the mask algorithm, which produces a third image, **$image3**. The final steps are to close $image2 (**CloseFrame $image2**) and $image3 (**CloseFrame $image3**).



The script is now recording. Your actions will appear below.

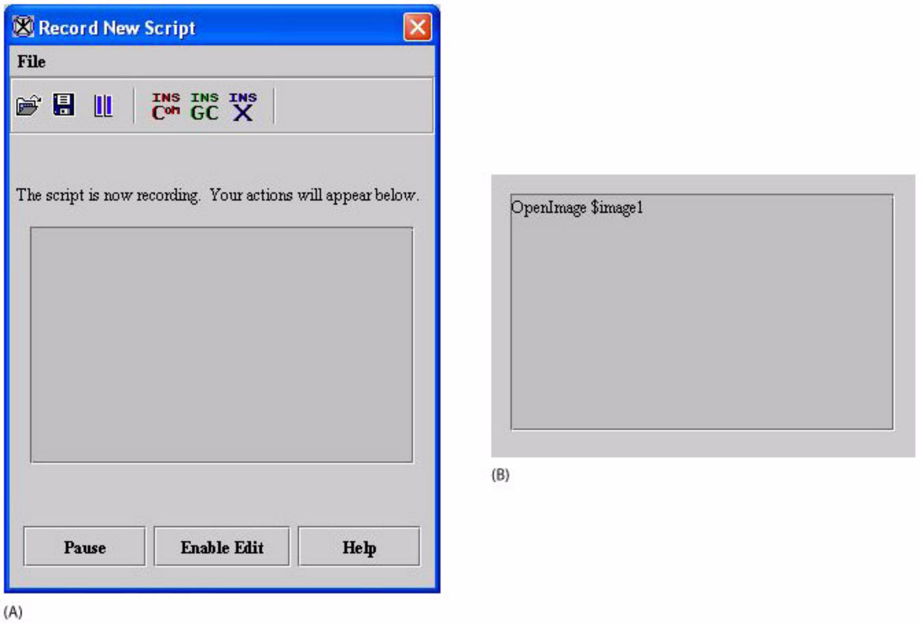
```
Clone $active1 $image1
AddMargins $image1 $image2 20 20 20 20 0 0 1.0
Mask $image2 $image3 false 0.0
CloseFrame $image2
CloseFrame $image3
```

To create scripts using the group mode

1. Start MIPAV. The initial MIPAV window and the Output window open.
2. Select File > Scripts > Record Script. The Record New Script dialog box (Figure 289A) opens.

   The following appears at the top of the dialog box: "The script is now recording. Your actions will appear below."
3. Open one or more images on which you want to run the script. The MIPAV window expands.

The command ***OpenImage $image1*** now appears in the scripting box (Figure 289B) in the Record New Script dialog box. If you continue to open images, the *$imageN* variable is sequentially ordered. That is, the command to open the second image is **OpenImage $image2**; the command to open the third image is **OpenImage $image3**; and so on.



The script is now recording. Your actions will appear below.

OpenImage $image1

Figure 289. The (A) Record New Script dialog box and (B) the scripting dialog box after opening an image in group mode

Note: Just as when you're creating a script in active mode, the Start recording script icon is red (●) to indicate that MIPAV is now recording a script.

4 Open any previously created VOI.

5 Perform any algorithms or utilities that you want recorded in the script.

6 Close the images.

7 Select File > Save in the Record New Script dialog box when you are finished creating the script. The Save dialog box opens.

8 Type a name for the script in File name.

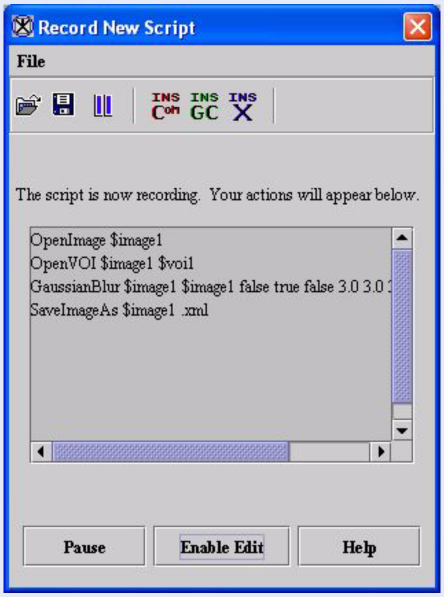
Tip: The file name extension for script files is either .sct or .xml. Make sure that you type one of these extensions at the end of the file name.

9 Click Save. The program saves the script in the scripts home directory.

10 Click File > Exit to close the Record New Script dialog box. The dialog box closes. You should now be able to select the script in Current Script.

An Example: Recording a Script in Group Mode

In this example, the user wants to open an image, open an associated VOI, run the Gaussian blur algorithm (replacing the image rather than creating a new image), and save and close the resulting image.



The script is now recording. Your actions will appear below.

```
OpenImage $image1
OpenVOI $image1 $voi1
GaussianBlur $image1 $image1 false true false 3.0 3.0 1
SaveImageAs $image1 .xml
```

To stop recording scripts during creation

If you are interrupted or need to take a break while you are recording a script, the Pause icon, ⏸, allows you to stop recording the script. When you click ⏸, the following message appears at the top of the Record New Script dialog box: “The script is now paused. Press Resume to resume recording." Notice that ⏸ changed to the Resume Scripting icon, ●.

When you can return to recording the script, click ●.

Running scripts

Depending on your preference, you can run scripts in active mode—from the scripting toolbar in the MIPAV window; or in group mode—on multiple images from the File menu in group mode.

To run scripts from the scripting toolbar (active mode)

1 Select a script from Current Script in the scripting toolbar.

Note: Current Script lists all of the scripts in the scripts home directory alphabetically.

2 Click ▶, the Run the selected script icon, on the scripting toolbar in the MIPAV window.

MIPAV automatically performs all of the actions in the script on the images indicated in the script.

Note: To reorder the list of scripts in Current Script, click 🔄, the Refresh icon.

To run scripts from the File menu (group mode)

1 Select File > Scripts > Run Script in the MIPAV window.

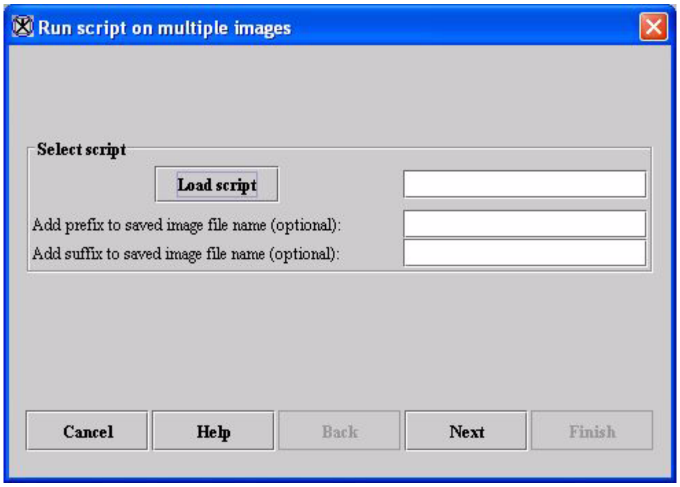
The Run Script on Multiple Images dialog box (Figure 290) opens.

2 Type, as an option, the prefix or the suffix that should appear in front of the file name.

3 Click Load Scripts. The Open dialog box opens showing all of the scripts in the scripts home directory.

4 Select a script.

5 Click Open.

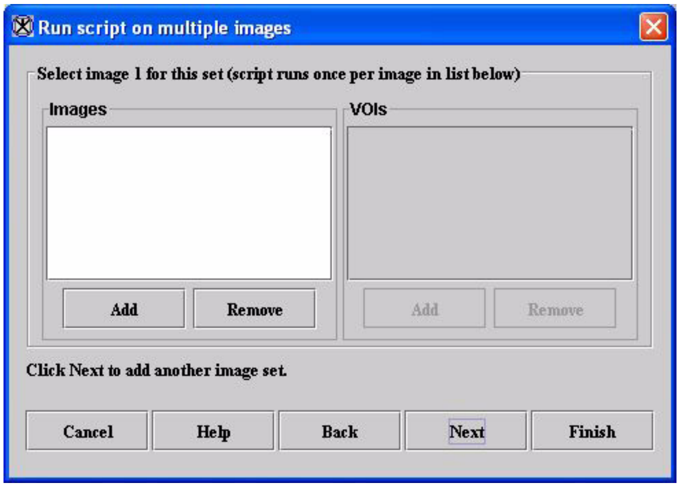

Run script on multiple images
Select script
Load script
Add prefix to saved image file name (optional):
Add suffix to saved image file name (optional):
Cancel
Help
Back
Next
Finish

Load script	Allows you to select the script that you want to run from the scripts home directory. When you select this button, the Open dialog box appears.
Add prefix to saved image file name	Specifies an alphanumeric prefix that you want to apply to the images produced by the script.
Add suffix to saved image file name	Specifies an alphanumeric suffix that you want to apply to the images produced by the script.
Cancel	Disregards any changes that you made in the dialog box and closes this dialog box.
Help	Displays online help for this dialog box.
Back	Returns to the previous dialog box without running the script.
Next	Proceeds to the next step in running the script.
Finish	Begins running the script.

Figure 290. Run Script on Multiple Images dialog box

6 Click Next. The Run Script on Multiple Images dialog box (Figure 291 on page 442) opens.

7 Click Add under the Images box on the left to add the image on which you want to run the script. The Open dialog box appears.



Select image 1 for this set (script runs once per image in list below)

Click Next to add another image set.

Images	Lists the images that appear in the script.
Add	Allows you to add one or more images on which to run the script.
Remove	Allows you to remove one or more images from those on which you plan to run the script.
VOIs	Lists any saved VOIs that are necessary when you run the script. This box is only enabled if the OpenVOI command was recorded in the script.*
Add	Allows you to add one or more VOIs that the script needs to run correctly. This button is only available if the OpenVOI command was recorded in the script.
Remove	Allows you to remove one or more VOIs from use with the script. This button is only available if VOIs are listed in the VOI box.
Cancel	Disregards any changes that you made in the dialog box and closes this dialog box.
Back	Returns to the previous dialog box without running the script.
Next	Proceeds to the next step in running the script.
Finish	Begins running the script.

*Each image can have a unique associated VOI, or a single VOI can be associated with all of the images in the script.

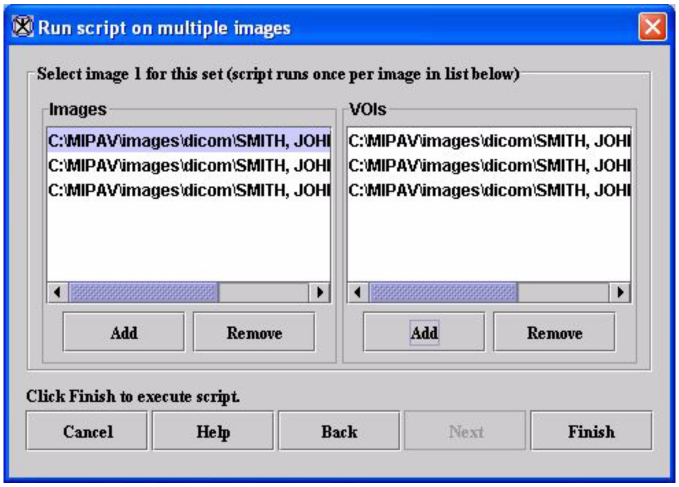
Figure 291. Run Script on Multiple Images dialog box (image and VOI selection)

8 Select one or more images, and click Open. The selected images appear in the Images box.

9 Select Add under VOIs to add one or more VOIs for that image. The Open dialog box appears.

10 Select an image in the Images box, and then select a VOI in the VOIs box to associate it with the image.

11 Repeat step 9 and step 10 for as many images and VOIs on which you want to run the script.



Run script on multiple images

Select image 1 for this set (script runs once per image in list below)

Images

C:\MIPAV\images\dicom\SMITH, JOHI
C:\MIPAV\images\dicom\SMITH, JOHI
C:\MIPAV\images\dicom\SMITH, JOHI

VOIs

C:\MIPAV\images\dicom\SMITH, JOHI
C:\MIPAV\images\dicom\SMITH, JOHI
C:\MIPAV\images\dicom\SMITH, JOHI

Click Finish to execute script.

Figure 292. Associating images and VOIs on the Run Script on Multiple Images dialog box

Tip: To speed the process of associating VOIs with images, you can open all of the images at once and all of their associated VOIs at once. The Images box shows all of the images selected and the VOI box lists all of the VOIs selected. Select an image in the Images box and then select the VOI that applies to that image. Repeat this process for each of the images.

12 Click Finish.

MIPAV automatically performs all of the actions in the script on the images indicated in the script.

Editing and deleting scripts

You can either edit a script while you are creating it, or you can edit a previously saved script. MIPAV saves the script in text format so that any text editor, such as Microsoft WordPad or NotePad, can modify the file.

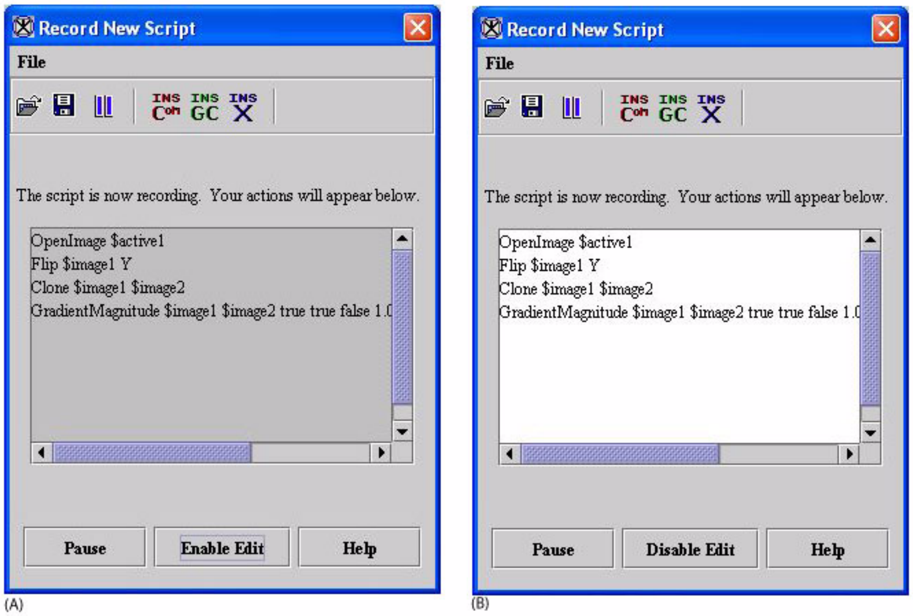
Caution: Editing scripts may cause serious problems if the script commands are not entered correctly. This task is recommended for more advanced users.

To edit scripts during creation

Suppose you make an error while you are recording a script, or perhaps you change your mind about performing a specific action. It can be more practical to correct the problem right away.

To correct errors or make changes to scripts while you're creating them, simply click Enable Edit, the Enable Edit button, in the Record New Script dialog box (Figure 293). Two things occur:

1. The scripting box in the dialog box turns from gray to white to indicate that you can now type, copy, or paste information into it and delete information from it.
2. The Enable Edit button changes to Disable Edit, the Disable Edit button. If the scripting box contains actions that you want to erase, select the actions and delete them. To add new actions, you can type them into the scripting box directly. You can also rearrange actions by copying and pasting them.



The script is now recording. Your actions will appear below.

```
OpenImage $active1
Flip $image1 Y
Clone $image1 $image2
GradientMagnitude $image1 $image2 true true false 1.0
```

Figure 293. Record New Script dialog box (A) before and (B) after clicking Enable Edit

When you are finished changing the script, click Disable Edit. The scripting box turns gray and the Disable Edit button becomes the Enable Edit button.

To edit previously created scripts

1 Click ●, the Start Recording Script icon, on the scripting toolbar. The icon turns red, and the Record New Script dialog box opens.

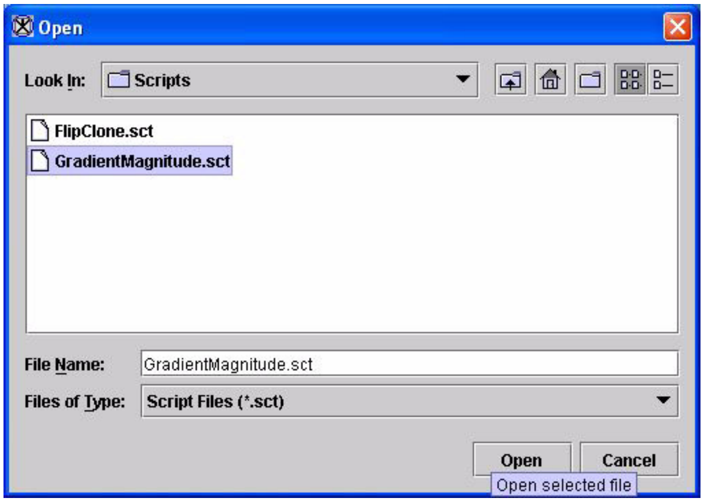
2 Select File > Open on the Record New Script dialog box. The Open dialog box (Figure 294) opens.

3 Select the script that you want to edit. The script appears in the scripting box on the Record New Script dialog box.

4 Click Enable Edit. The actions box turns from gray to white.

5 Select any of the actions and copy, rearrange, or delete them, or type new actions directly into the scripting box.

6 Click Disable Edit when you are satisfied with the script. The scripting box returns to gray.



File Name: GradientMagnitude.sct

Files of Type: Script Files (*.sct)

Figure 294. Open dialog box showing scripts in the scripts home directory

7 Continue recording actions if needed until you are satisfied the script.

8 Save the script by clicking File > Save. The Save dialog box opens.

9 Type the name of the script followed by the .sct extension in the File Name box.

10 Click Save. MIPAV saves the script under the specified name.

To delete scripts

If you decide at some point that you want to delete a script, go to Windows Explorer and navigate to the scripts home directory.

1 Right-click Start on the Windows desktop.

2 Select Explore. The Windows explorer window opens.

3 Navigate to the scripts home directory.

4 Select the script or scripts that you want to delete.

5 Right-click on the selected script or scripts.

6 Click Delete. Windows deletes the selected scripts and places them in the Recycle Bin where they stay until you empty the bin.

Combining scripts and other programs

To increase productivity and efficiency, you can integrate MIPAV functions into your normal workflow by calling the scripts you've created. A simple example is creating a DOS batch file that opens and runs MIPAV scripts.

Using the mipav command

To call scripts from other programs, you use the **mipav** command in the Command Prompt dialog box. The correct syntax of this command follows.

Syntax of the mipav command

```
mipav [-hH] [-iI] imageFileName [-sS] ScriptFileName [-vV] voiFileName [-hideHide]
```

Parameters	Purpose
[-h][-H]	Displays help for the **mipav** command in a Command Prompt window
[-hide][-HIDE]	Hides application frame
[-i][-I]	Image file name
[-s][-S]	Script file name
[-x][-X]	XML script file name
[-v][-V]	VOI file name
[-o][-O]	Specifies the output file name when “Save Image As” script command is used
Exit	Exits the MIPAV program

Figure 295 shows examples of the **mipav** command.

Note: When calling MIPAV scripts from other programs, be sure to enter an Exit command at the end of the MIPAV script.

Example 1: Starts MIPAV

```
> mipav
```

Example 2: Starts MIPAV and opening an image

```
> mipav imageFileName
```

Example 3: Starts MIPAV but does not display frame, opens an image, and runs a script on the image.

```
> mipav -i imageFileName -s scriptFileName -hide
```

Example 4: Starts MIPAV, runs a script, opens the first image, opens two VOIs associated with that image, opens a second image, and associates another VOI with that image

```
> mipav -s scriptFileName -i imageFileName1 -v voiName1 -v voiName2 -i imageFileName2
-v voiName3
```

Example 5: Exiting the MIPAV program

```
> mipav Exit
```

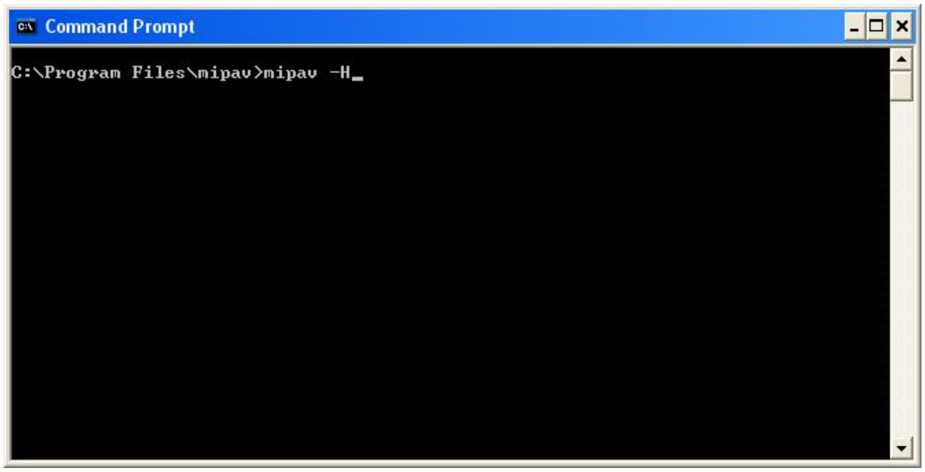
Figure 295. Examples of using the mipav command

To display help for using the mipav command

1 Navigate to the mipav directory on your computer.

2 Select Start > All Programs > Accessories > Command Prompt. The Command Prompt dialog box opens.

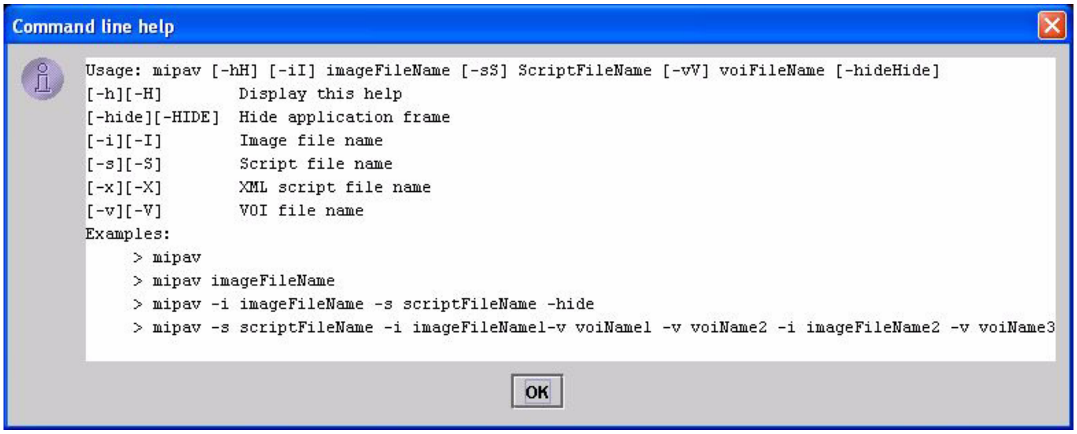
3 Type **mipav -H** (refer to Figure 296).



```
C:\Program Files\mipav>mipav -H_
```

Figure 296. Command Prompt dialog box showing command to open Command Line Help dialog box

4 Press Enter. The Command Line Help dialog box (Figure 297) opens.



```
Usage: mipav [-hH] [-iI] imageFileName [-sS] ScriptFileName [-vV] voiFileName [-hideHide]
[-h][-H]        Display this help
[-hide][-HIDE]  Hide application frame
[-i][-I]        Image file name
[-s][-S]        Script file name
[-x][-X]        XML script file name
[-v][-V]        VOI file name
Examples:
       > mipav
       > mipav imageFileName
       > mipav -i imageFileName -s scriptFileName -hide
       > mipav -s scriptFileName -i imageFileName1-v voiName1 -v voiName2 -i imageFileName2 -v voiName3
```

Figure 297. Command Line Help dialog box, which shows the syntax of the mipav command as well as examples

To open a DICOM image dataset

Suppose you want to open a single DICOM image from a collection of experiments made in 2004 named exp2004. You would type the following command in the Command Prompt dialog box in Microsoft Windows XP:

```
C:\ mipav -i i:\images\DICOM\exp2004\I04301.dcm
```

To open VOIs into that image

You can open VOIs as well as the image from the command line. In Windows XP, it would be the following:

```
C> mipav -i i:\images\DICOM\exp2004\I04301.dcm -v i:\VOIs\exp2004\levelset1.xml
```

In a UNIX BASH shell, this command looks like:

```
$ mipav -i ~/images/DICOM/exp2004/I043401.dcm -v ~/VOIs/exp2004/levelset1.xml
```

To open multiple images using compound commands

Suppose you know that there were multiple DICOM datasets in exp2004. To open every DICOM image on the Windows computer, you would type:

```
C> for %f in (i:\images\DICOM\exp2004\*01.dcm) do mipav -i %f
```

In this case, you must know something of the file structure of that dataset—you assumed that all image datasets had only one image ending in 01. However, the disadvantages of this format is the possibility of not opening all of the images at the same time.

A similar loop to open image sets on a UNIX BASH command line looks like:

```
$ for FI in `ls ~/images/DICOM/exp200?/*01.dcm`; do ./mipav -i $FI &; done
```

There are three significant differences between the BASH command and the Windows command (beside from how a directory is specified):

- **The use of the ls command when listing the directory—**The reason you must list the ls is due to the way a for loop works in BASH. The **for** requires a command and uses that command's return value as the boolean test to continue repeating the interior list of commands. By contrast, the Windows command shell expects a list of files. So long as the file listing has more results to list, BASH continues to repeat the **mipav** command.

- **The use of a wildcard when listing the directory**—BASH allows the directory list to use wildcard characters in more than one location, which permits searching for the images in any seven-character directory beginning with exp200 as well as all files ending in 01.dcm. This means that MIPAV starts with images from the exp2004 directory, as well as exp2003 or should it exist, exp200M, since the ? matches any character, not just a number. This is an example of a feature of the shell being used to expand the results. Windows command shell does not support this feature.
- **Sending the mipav command to operate in the background**—BASH is a shell that allows *job control*. Using this feature allows you to start MIPAV and continue it asynchronously, permitting BASH to retain control. BASH can then continue processing the loop and starting MIPAV with the next matching file. Each MIPAV runs concurrently and allows you to manipulate each image with MIPAV at will. Although this allows you to see the images at the same time, the disadvantage is that the various windows begin to clutter the screen causing operator confusion.

When there is more than one MIPAV application window running, it’s possible to close the wrong image by closing the wrong MIPAV application. In addition, operations that can occur between windows when running a single MIPAV may not be transferable between images being run by separate MIPAV windows.

While starting more than one MIPAV to display a set of images may be fine in limited applications, it causes needless overhead within in the operating system wasting system resources.

Using Shell scripting to lessen typing

Using shell scripts to reduce the amount of repetitive work is a common reason for writing a script. When best used, several small scripts that work in concert can reduce the amount of typing required and the amount of time needed and can automate tasks.

The following example uses a Windows command shell to illustrate how you can shorten the number of keystrokes required. In this case, you would write a batch file to load a levelset VOI into an image.

```
@echo off
rem -- %1 is the full path to the image file, though not
rem -- the file itself; we assume there to be a *01.dcm
rem -- file to exist in this directory.

rem -- VOI is assumed to be in the same directory with name
rem -- levelset1.xml

./mipav -i %1\*01.dcm -v %1\levelset1.xml
```

More efficient and more useful, starting MIPAV with multiple images is easily done in a simple script. Here is how it is done in BASH:

```
#! /bin/bash
# argument 1 is the file (with wild-cards) we want to open
# arg 1 must be escaped (with quotes) to allow the shell to send
# the wildcards unmolested to the script. Otherwise, the shell
# will try to expand the shorthand. This has a different effect.

LISTING=`ls $1`  # Generate the file listing
MIPARGS=
  # For each file in the listing, prepend it with '-i' and
  # the filename, then follow it with all the previous
  # files.
for FS in $LISTING;
 do
 MIPARGS=" -i $FS $MIPARGS";
done

 # start MIPAV:
./mipav $MIPARGS
```

Although this script doesn't include the line `./mipav $MIPARGS` with a "&" to run MIPAV in the background, it could have. This would have the effect of exiting the script with MIPAV in the background; as it is, the script does not exit—and return control to you at the command line—until MIPAV exits.